A Method for Simplifying Boolean Functions

By A. H. SCHEINMAN

(Manuscript received March 6, 1962)

This article presents an iterative technique for simplifying Boolean functions. The method enables the user to obtain prime implicants by simple operations on a set of decimal numbers which describe the function. This technique may be used for functions of any number of variables.

At the present time, although several design aids have been introduced,[1,2,3] the synthesis of switching circuits remains a highly developed art, the theory being of only limited value to the circuit designer. In particular, that part of the design process involving the simplification of Boolean functions having large numbers of variables still presents a major problem.

Probably the best method currently available for the solution of such problems is the Quine-McCluskey Tabular Method, which consists of the exhaustive comparison of terms of the standard sum for adjacencies — terms which differ in only one variable. This technique, besides being a long, tedious one, determines all possible prime implicants of the given function. Thus a second problem is generated — selecting the essential prime implicants from among those found. This in itself is often a difficult procedure, for which specific methods have been developed.[1]

The method to be described in this paper is a simple, iterative technique for determining the prime implicants of a Boolean function. All of the essential prime implicants are found with this method, and in general, some or all of the nonessential prime implicants are automatically eliminated, materially simplifying the final search for the essential prime implicants.

Any Boolean function, say $f(x_1, \cdots, x_n)$ for example, may be expanded into the form

$$f(x_1, x_2, \cdots, x_n) = x_1 \cdot f(1, x_2, \cdots, x_n) + x_1' \cdot f(0, x_2, \cdots, x_n).$$

The above theorem is generally referred to as the expansion theorem[4] and enables one to expand any n-variable switching function about any one of the variables. As shown above the given function is said to be expanded about the variable x_1. Two new functions are thus formed, one multiplying x_1 and one multiplying x_1'. The two new switching functions are functions of $x_2, \cdots, x_n$ only, and are referred to as the residues of x_1.

Let us assume that the given function is specified in its canonical form as a sum of product terms,

$$f(x_1, x_2, \cdots, x_n) = \sum P_j$$

where P_j represents the general product term and the subscript j is the decimal equivalent of the associated product term, when the product terms are expressed in binary form with primed literals replaced by zeros and unprimed literals replaced by ones. This is often written as

$$f(x_1, x_2, \cdots, x_n) = \sum j, \qquad P \text{ being implied.}$$

Assume further that the variables are assigned binary weights in the order shown, with x_1 being assigned the highest weight and x_n the lowest. When the variable weights are assigned in this manner the expansion becomes a very simple process. In particular, if a specific series of decimal numbers characterizes the function, the residues may be formed as follows:

$$f(x_1, x_2, \cdots, x_n) = \sum j = x_1 R_1 + x_1' R_2 .$$

R_2 is a sum consisting of the decimal numbers which are smaller than the weight of x_1 (the variable being expanded about), and R_1 is a sum consisting of the numbers which are greater than the weight of x_1, from each of which the weight of x_1 is subtracted.

Noting that the residues R_1 and R_2 are now each specified by summations, and the new sets of decimal numbers are interpreted as being functions of all succeeding variables, generically we may write:

$$R = f(x_2, x_3, \cdots, x_n).$$

Each of the residues can now be expanded about x_2, which is the highest weighted variable of the residues. Thus the expansion can be carried out in this manner using nothing more complicated than the subtraction process.

As an example, assume the following function:

$$f(A,B,C,D) = \sum(0,1,8,9,10) = AR_1 + A'R_2$$

where the order of the variables (A,B,C,D) indicates the relative binary weights, i.e., A is the highest weighted and D is the lowest weighted variable.

$$R_2 = f(B,C,D) = \sum 0,1 \quad \text{and}$$

$$R_1 = f(B,C,D) = \sum(8\text{-}8),\ (9\text{-}8),\ (10\text{-}8) = \sum 0,1,2$$

since the binary weight of $A = 8$.

This process may be repeated by expanding each of the residues about B. At this point let us examine R_1 and R_2.

There are five possibilities:

(a) $R_1 = R_2 = R$, therefore

$$f(A,B,C,D) = AR + A'R = R$$

indicating that A and A' are redundant

(b) $R_1 > R_2$, indicating that the decimal numbers representing R_2 form a subset of those representing R_1, then $R_1 = R_2 + R_b$, and

$$f(A,B,C,D) = A(R_2 + R_b) + A'R_2 = AR_2 + AR_b + A'R_2 = R_2 + AR_b .$$

(c) $R_2 > R_1$, where $R_2 = R_1 + R_a$

$$f(A,B,C,D) = AR_1 + A'(R_1 + R_a) =$$

$$AR_1 + A'R_1 + A'R_a = R_1 + A'R_a .$$

(d) The residues have no numbers in common, in which case

$$f(A,B,C,D) = AR_1 + A'R_2 .$$

(e) Some of the numbers in each residue are the same,

$$R_1 = R_c + R_d , \qquad R_2 = R_c + R_e \quad \text{and}$$

$$\begin{aligned} f(A,B,C,D) &= A(R_c + R_d) + A'(R_c + R_e) \\ &= AR_c + AR_d + A'R_c + A'R_e \\ &= R_c + AR_d + A'R_e . \end{aligned}$$

Note that in this case the function may be written as

$$f(A,B,C,D) = R_c + A(R_c + R_d) + A'(R_c + R_e)$$

The summation R_c may be included with A and A' as a redundancy. This is significant in the method to be shown, because R_c may contribute to the simplification of residues resulting from subsequent

expansion. This is analogous to using a term of the standard sum in more than one subcube when using a Karnaugh map.[2]

The following method employs all of the above ideas and will be shown by means of an example.

Assume a function

$$f(A,B,C,D) = \sum 1,2,3,4,6,7,8,9,11,12,13,14.$$

Expansion about the variable A — the highest weighted variable — yields

$$\begin{aligned} f(A,B,C,D) &= A'(\textstyle\sum 1,2,3,4,6,7) + A(\textstyle\sum 0,1,3,4,5,6) \\ &= \textstyle\sum 1,3,4,6 + A'(\textstyle\sum 1,2,3,4,6,7) + A(\textstyle\sum 0,1,3,4,5,6). \end{aligned}$$

Where the summation $\sum 1,3,4,6$ is that part of the A residues which are the same, note that the numbers 1,3,4,6 are now redundant in the summations multiplying A and A'. The summation 1,3,4,6 actually represents those terms of the standard sum which differ only in the A variable.

There are now three summations, each of which is a function of B,C,D only. Each of these functions must now be expanded about the variable B. These operations are repeated exactly until the expansion about all variables is complete.

At this point the above illustration will be repeated, showing a mechanical technique to organize and simplify the procedure. Refer to Fig. 1.

Step 1. Arrange the decimal numbers representing the given function in a column.

Step 2. Divide the decimal numbers into two columnar groups, one headed with A' and one headed with A. The A' column contains the numbers of the original function which are smaller than 8 — the binary weight of A — and the A column contains the numbers which are equal to or greater than 8, first subtracting 8 from each.

Step 3. Include a third column, headed by a dash to indicate the redundancy of A and A', consisting of the numbers which are common to columns A and A'. Check the corresponding numbers in columns A and A' to record the fact that they are redundant. If any of the numbers in the dashed column have been *previously* checked in *both* the A and A' columns, they should also be checked in the dashed column.

Step 4. Examine each column. If any column consists of only checked numbers, eliminate the column entirely.

Each of the columns must now be expanded about B by repeating the above steps. The expansion of the function in the A' column is shown in

Fig. 1b. Since the weight of B is 4, all numbers less than 4 are placed in the B' column. These numbers are 1,2,3. Note that 1 and 3 must be checked since they were previously checked in the A' column. The numbers 4,6,7 are placed in the B column, first subtracting 4 from each, giving 0,2,3. The numbers 0,2 which correspond to 4,6 must be checked. A dashed column consisting of the numbers 2,3 is now included. Check the numbers 2,3 in both the B and B' columns. All of the numbers in the B and B' columns are now checked, hence both columns may be eliminated as shown.

Figure 2 illustrates the complete development. When the function is expanded about the final variable, note that the residues must be 0. At this point the prime implicants may be determined by simply tracing a path back to the start and reading the appropriate columnar headings.

Not all of the prime implicants obtained may be required to describe the function. In the example just shown, all of the prime implicants were essential, but another example will be shown in which this is not the case.

$$f = \sum 0,1,2,3,4,6,7,8,9,11,15$$

This function is simplified in Fig. 3, and four prime implicants are obtained, not all of which are essential. A simple method for determining

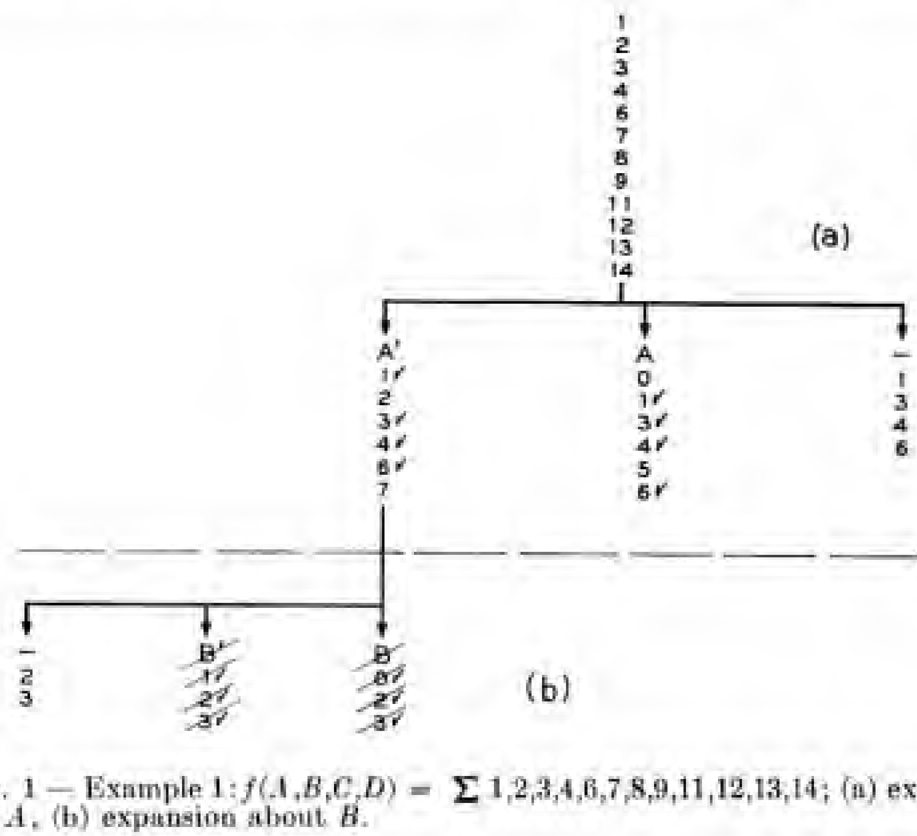
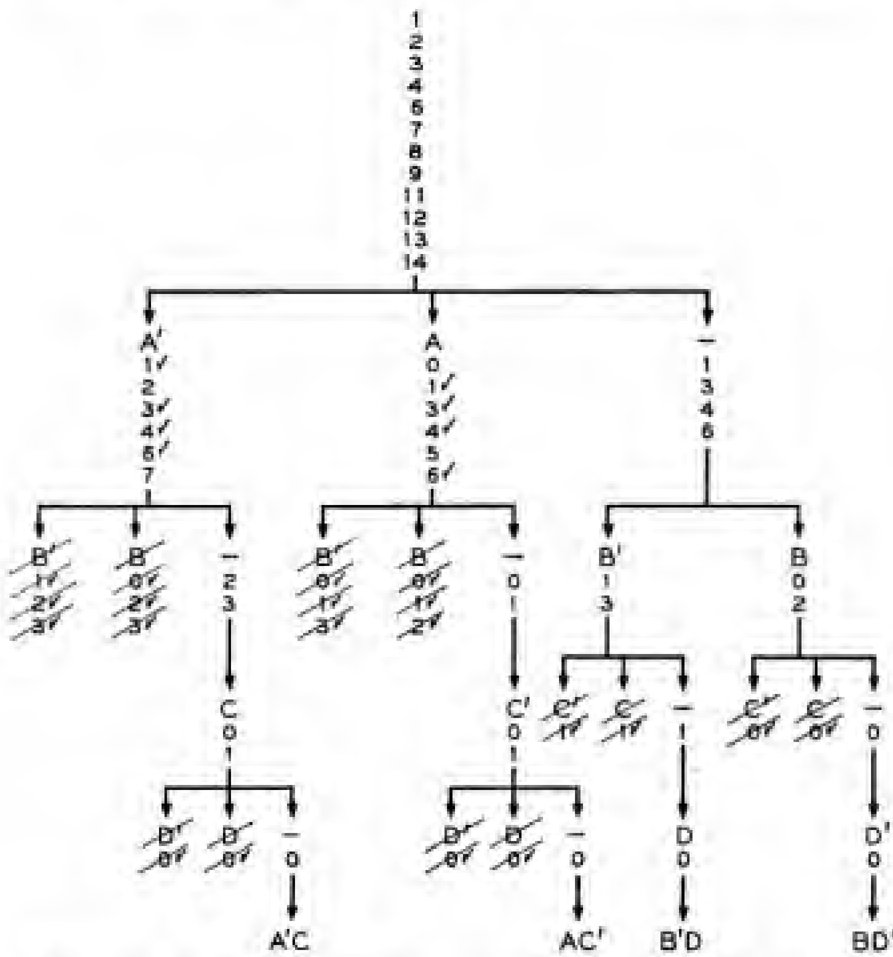


Fig. 1 — Example 1: $f(A,B,C,D) = \sum 1,2,3,4,6,7,8,9,11,12,13,14$; (a) expansion about A, (b) expansion about B.



$f_{(A,B,C,D,)} = \Sigma\ 1,2,3,4,6,7,8,9,11,12,13,14 = A'C + AC' + B'D + BD'$

Fig. 2 — Example 1 completed.

the essential prime implicants is available in a prime implicant chart first proposed by Quine and later simplified by McCluskey. Such a chart is shown in Fig. 4 for the example of Fig. 3 (see Ref. 1).

The chart requires the establishment of columns, each of which represents one of the decimal numbers of the original function and is so headed. Each row represents one of the prime implicants and is thus identified.

Each prime implicant is a combination of 2^k of the decimal numbers of the original function; k may be any integer, including zero. It is an easy matter to find these numbers. One could, for example, trace backward from the final residue of the prime implicant (always zero) and if a column is passed through which is headed by an unprimed variable add the weight of the variable, if through a primed column the number remains unchanged. When tracing through a column headed by a dash

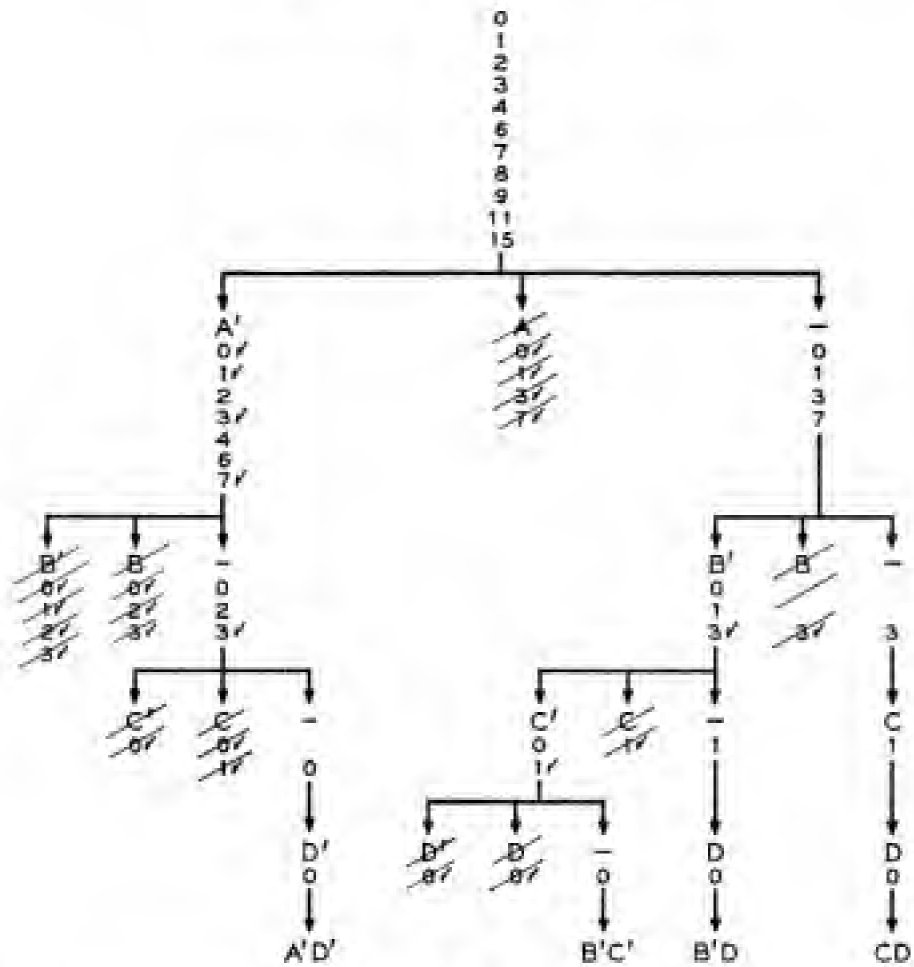


Fig. 3 — Example 2: $f(A,B,C,D) = \sum 0,1,2,3,4,6,7,8,9,11,15$; example yielding nonessential prime implicants.

each number becomes two numbers, one of which is the same as the number in the dashed column. The other number is obtained by adding the weight of the variable from whose expansion the dashed column resulted.

On each row of the chart, mark a cross under the decimal numbers

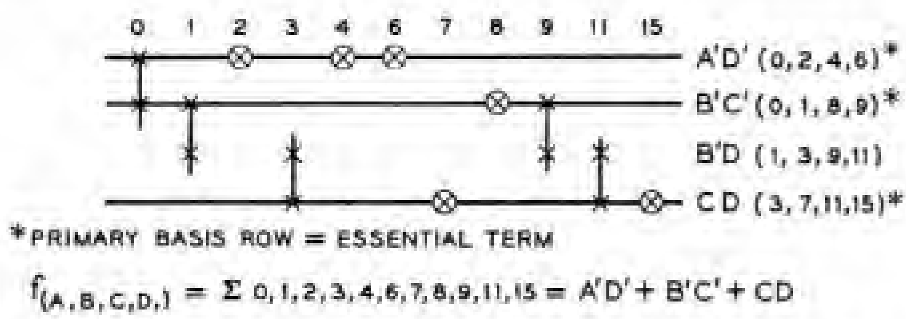


Fig. 4 — Chart method of obtaining essential prime implicants for Example 2.

associated with the terms contained in the prime implicant represented by that row. Then scan the columns and circle the crosses which stand alone in a column, and rule a line through each associated row. Such rows represent essential prime implicants. Now rule a vertical line through each cross in the ruled rows. If all other crosses in the chart are not thus ruled out by the vertical lines, additional prime implicants must be chosen. (See Ref. 1 for further discussion of prime implicant charts.) The chart of Fig. 4 shows that $A'D'$, $B'C'$, and CD are essential and the function may therefore be expressed by the following minimum sum

$$f(A,B,C,D) = A'D' + B'C' + CD.$$

If the function of Fig. 3 is simplified by other tabular methods, it will be noted that there are six possible prime implicants. This illustrates an important advantage of the method described here, and this is that while all essential prime implicants are obtained, some or all of the non-essential prime implicants may be eliminated automatically, materially simplifying the search for the essential terms by charting. The missing prime implicants are actually included in columns which were crossed out in the development because all elements were checked, indicating that the prime implicants if obtained would be redundant.

There are functions for which, if all possible prime implicants are found, charts would be produced which are cyclic in form. There are no immediately apparent choices of prime implicants which would yield a minimum sum. In such cases some initial choice must be made, and some cut and try is necessary. In fact, there will generally be more than one equally satisfactory solution, depending upon the initial choice.

When the method described in this paper is applied to functions which would normally produce a cyclical prime implicant chart, the chart obtained will not be cyclical. Some of the prime implicants will be eliminated as the work progresses, in effect making the initial choices automatically. This has two results. The chart is materially simplified and the solution is easily and automatically obtained. However if a particular initial choice would result in a more economical solution than another, then this method may or may not obtain the minimum sum, depending upon the weighting of the variables. The final solution in such cases must be regarded as an approximation to the minimum sum. The approximation will always be a close one.

If some of the decimal numbers specifying the given function are "don't care" terms, they must be checked initially and subsequently treated like any other checked number. The only real difference is noted when drawing a prime implicant chart, where columns corresponding to "don't care" terms would not be included (Refs. 1, 4).

A final example is included in Figure 5 to indicate which of the decimal numbers of the given function are combined to obtain any particular number in the expansion. The decimal numbers of the given function are shown parenthetically.

In conclusion the method described above offers the following advantages:

(a) The decimal numbers specifying the function may be operated on directly without any preliminary grouping.

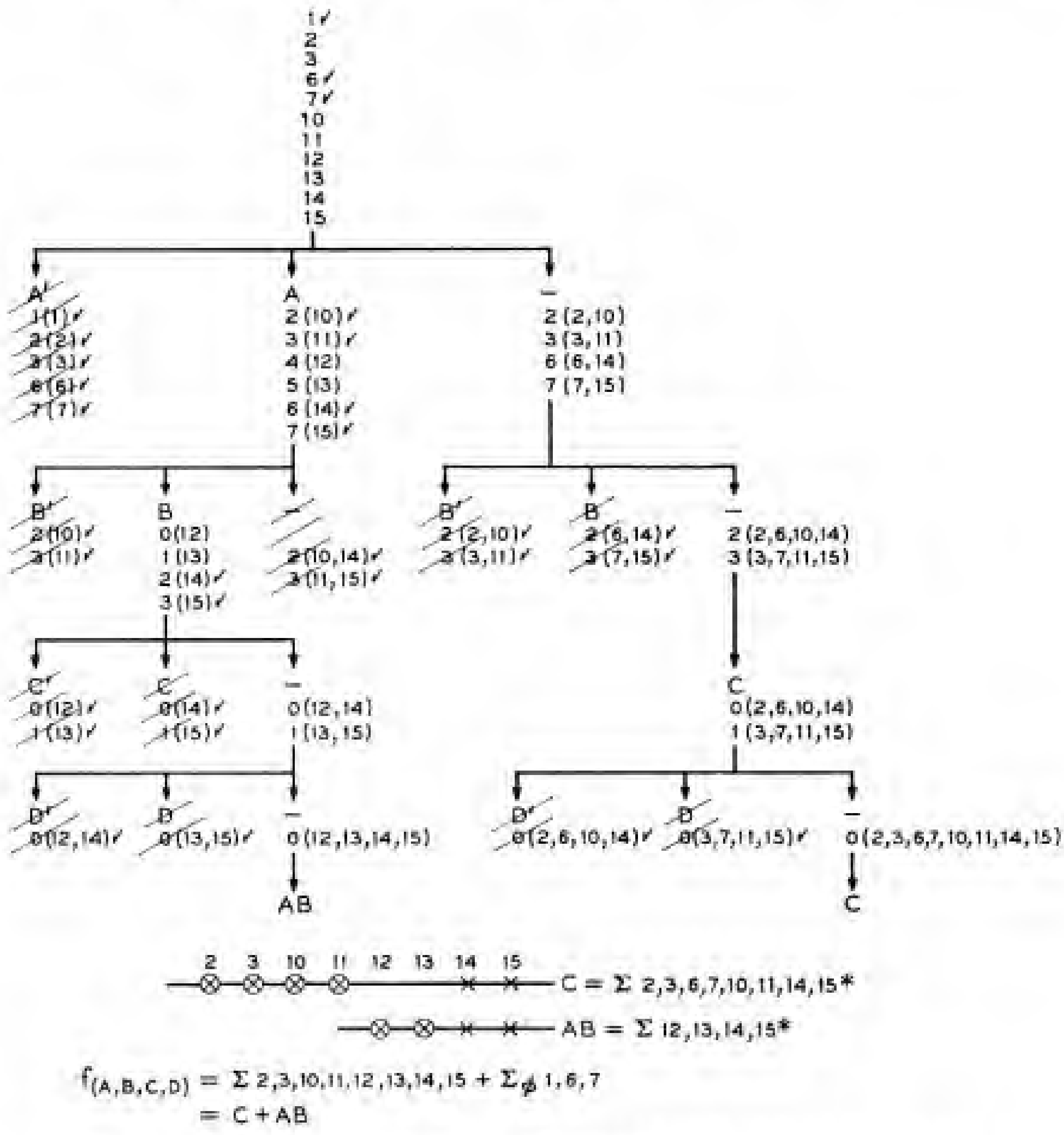


Fig. 5 — Example 3: $f(A,B,C,D) = \sum 1,2,3,6,7,10,11,12,13,14,15$; numbers in parentheses are decimal numbers which were combined to give numbers in expansion.

(b) Adjacencies (terms which differ in only one variable) are found in groups, making this method very rapid in use.

(c) The operations are simple and iterative, enabling functions having any number of variables to be simplified.

(d) Some nonessential prime implicants will in general be eliminated, simplifying the prime implicant chart.

REFERENCES

1. McCluskey, E., Jr., Minimization of Boolean Functions, B.S.T.J., **35**, 1956, pp. 1417–1445.
2. Karnaugh, M., The Map Method for Synthesis of Combinatorial Logic Circuits, A.I.E.E. Trans., Part 1: Communications and Electronics, **72**, 1953, pp. 593–599.
3. Hall, F. B., Boolean Prime Implicants by the Binary Sieve Method, A.I.E.E. Trans., Part 1: Communications and Electronics, 1962, pp. 709–713.
4. Caldwell, S. H., *Switching Circuits and Logical Design*, John Wiley & Sons Inc., N.Y., 1958.